pressure_sensor_noise_test

September 16, 2021

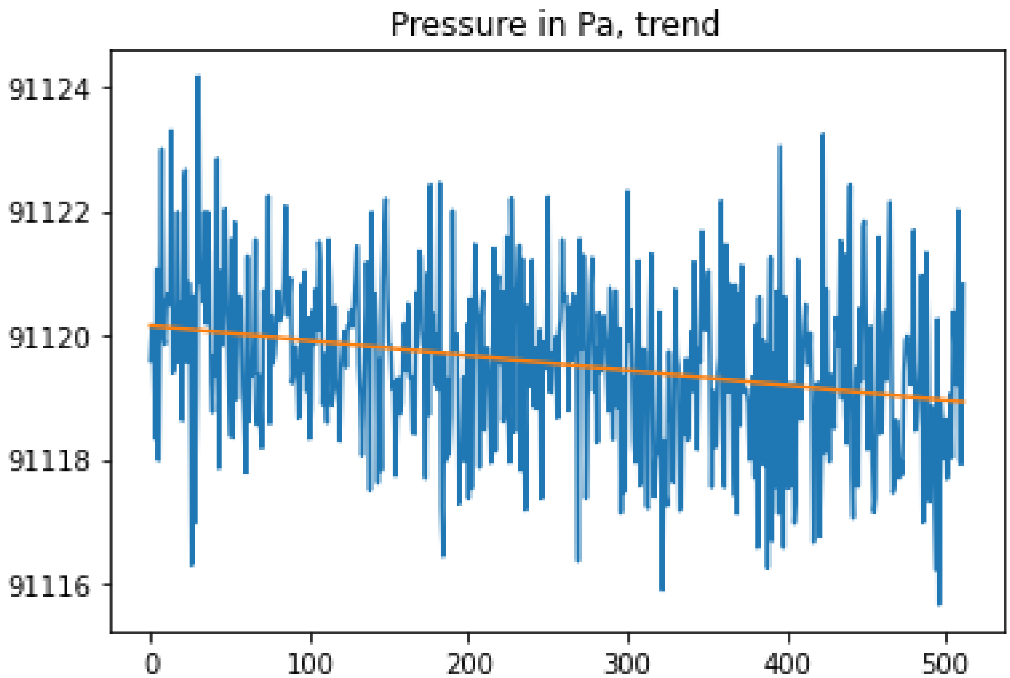
[1]:
```python
import csv
import statistics
from scipy.stats import normaltest
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
from scipy.stats import probplot
```

1 MS5611, 50Hz sample rate

[2]:
```python
pa = []
zcm = []
with open('ms5611_512samples_1.txt', newline='') as csvfile:
    csvreader = csv.reader(csvfile, delimiter=' ')
    for row in csvreader:
        pa.append(float(row[0]))
        zcm.append(float(row[1]))
pa_arr = np.array(pa)
zcm_arr = np.array(zcm)
pa = []
zcm = []
```

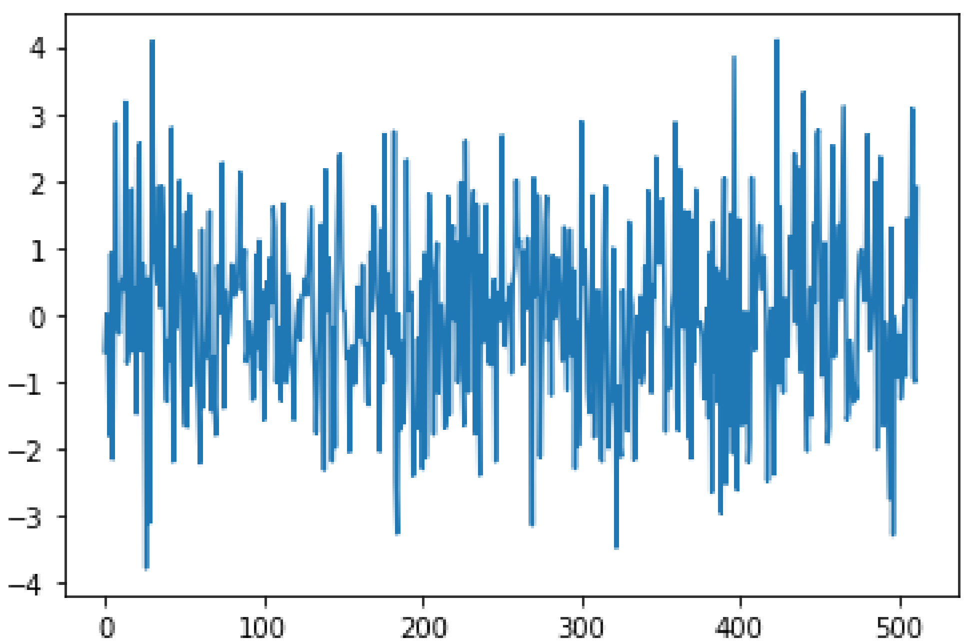
2 Pressure Statistics

[3]:
```python
X = list(range(0, len(pa_arr) ) )
X = np.reshape(X, (len(X), 1))
model = LinearRegression()
model.fit(X, pa_arr)
trend = model.predict(X)
# plot trend
plt.plot(pa_arr)
plt.title('Pressure in Pa, trend')
plt.plot(trend)
plt.show()
```



3 De-trend data

[4]:
```
pa_detrended = pa_arr - trend
plt.plot(pa_detrended)
plt.show()
```



[5]:

```
pa_variance = statistics.variance(pa_detrended)
print("Pressure noise variance = {0} Pa^2".format(pa_variance))
```

```
Pressure noise variance = 1.8235212584490543 Pa^2
```

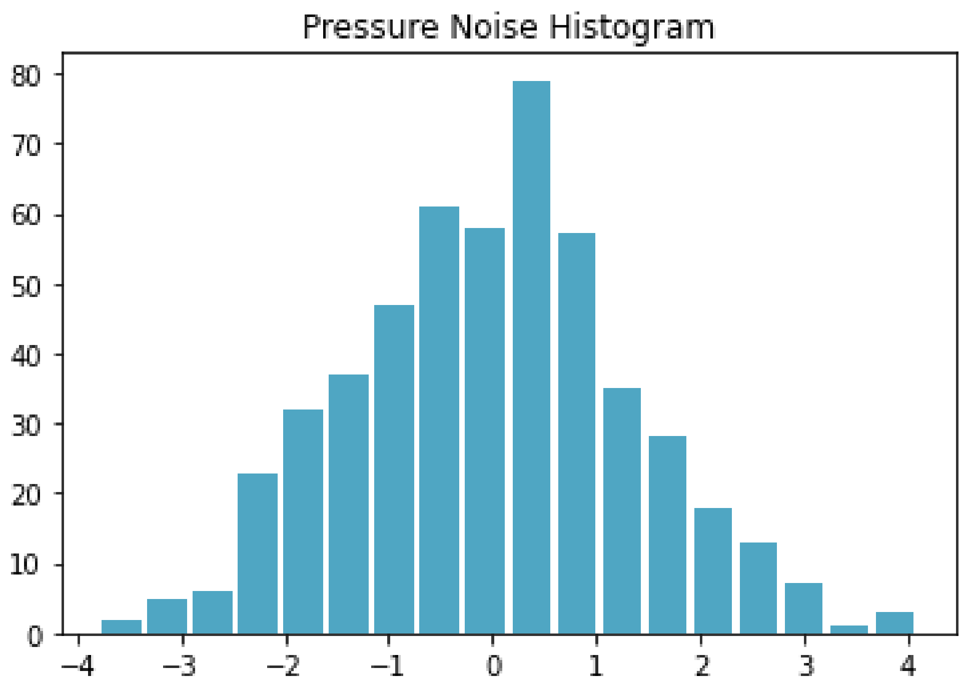
[6]:

```
statistic, p_value = normaltest(pa_detrended)
print("Pressure noise normality test : p-value = {0}".format(p_value))
```

```
Pressure noise normality test : p-value = 0.6581999509618583
```

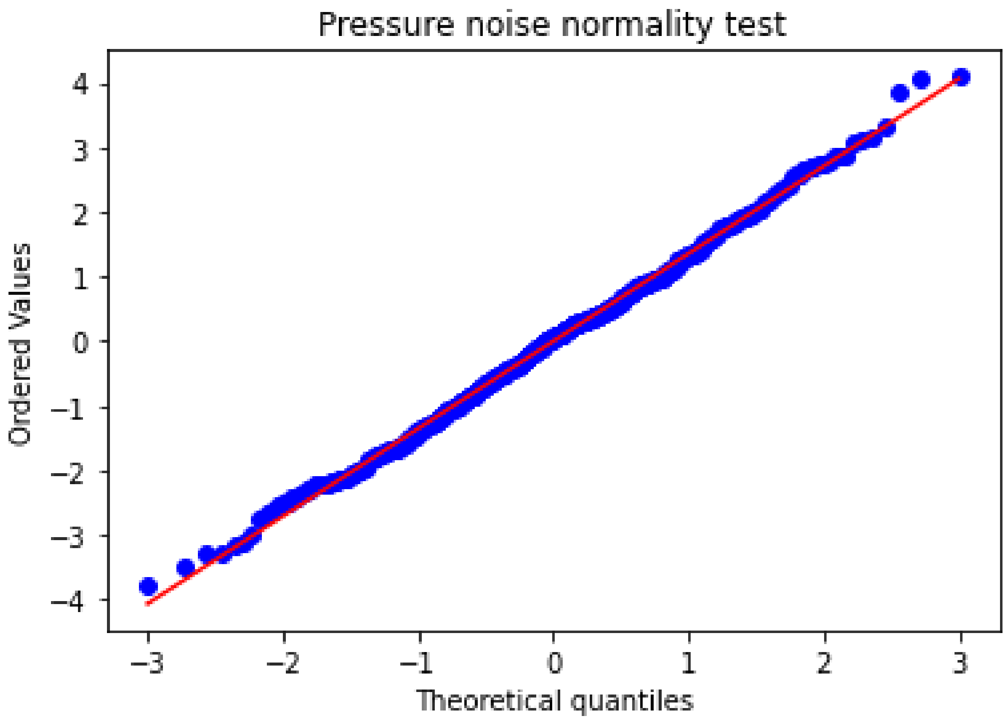
[7]:

```
n, bins, patches = plt.hist(x=pa_detrended, bins='auto', color='#0580aa',
                            alpha=0.7, rwidth=0.85)
plt.title('Pressure Noise Histogram')
plt.show()
```



[8]:

```
probplot(x=pa_detrended,dist='norm',plot=plt)
plt.title('Pressure noise normality test')
plt.show()
```



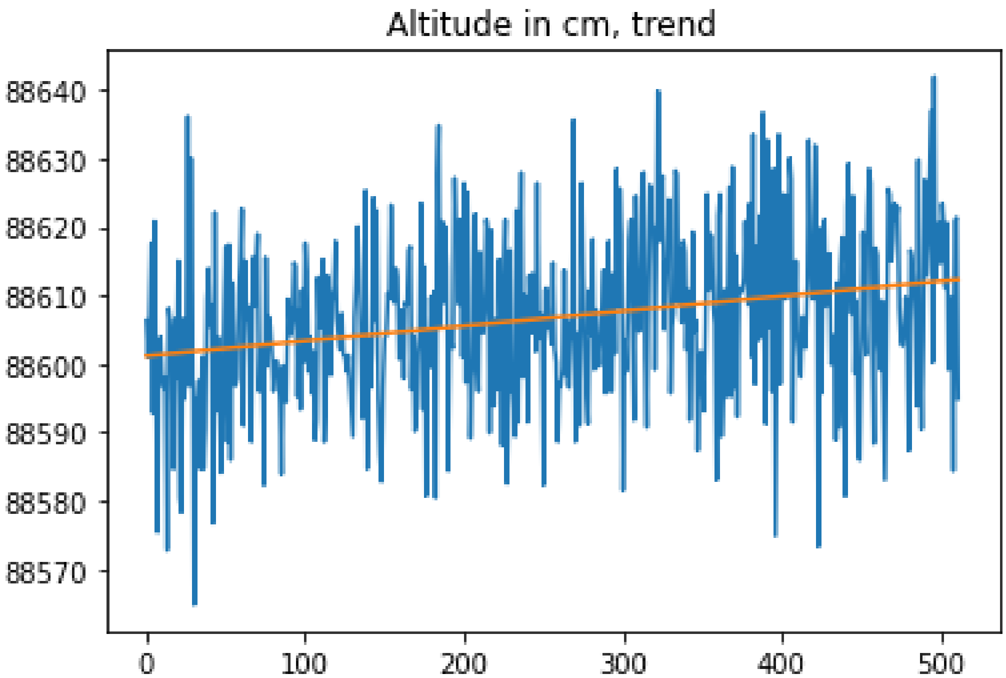
4 Altitude Noise Statistics

On a macro scale, the pressure - altitude relationship is highly non-linear

`Pressure Altitude (in cm) = 4430769.396f * (1.0f - pow(pa/101325.0f, 0.190284f))`

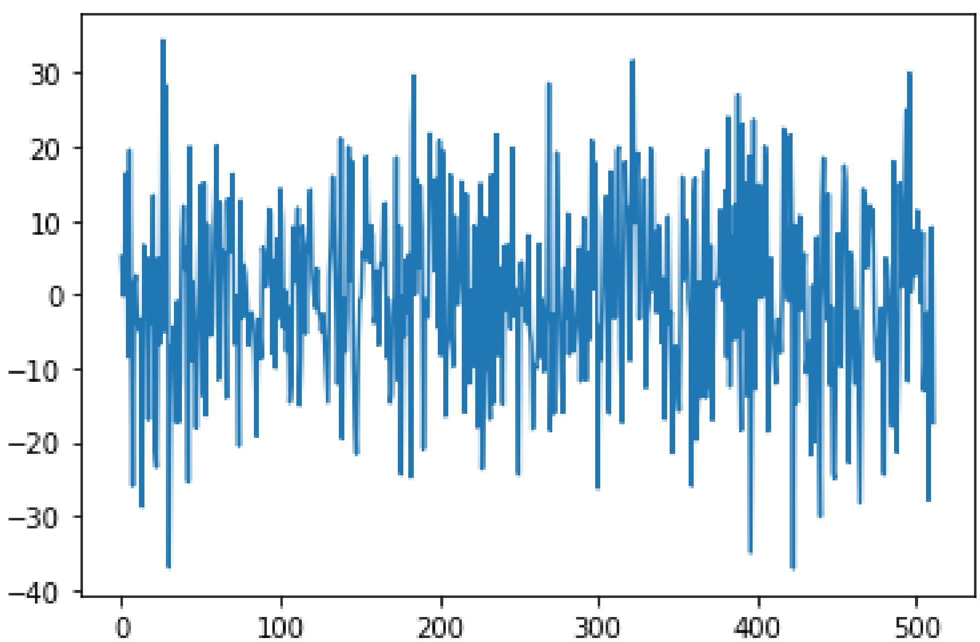
where the pressure is in Pascals (Pa) and sea-level pressure is nominally 101325Pa. However, when you consider a neighbourhood of +/-50 Pascals, the relationship is linear enough so that altitude measurements from a barometric pressure sensor also have a Gaussian distribution for noise.

[9]:
```
model = LinearRegression()
model.fit(X, zcm_arr)
trend = model.predict(X)
# plot trend
plt.plot(zcm_arr)
plt.title('Altitude in cm, trend')
plt.plot(trend)
plt.show()
```



5 De-trend data

[10]:
```
zcm_detrended = zcm_arr - trend
plt.plot(zcm_detrended)
plt.show()
```



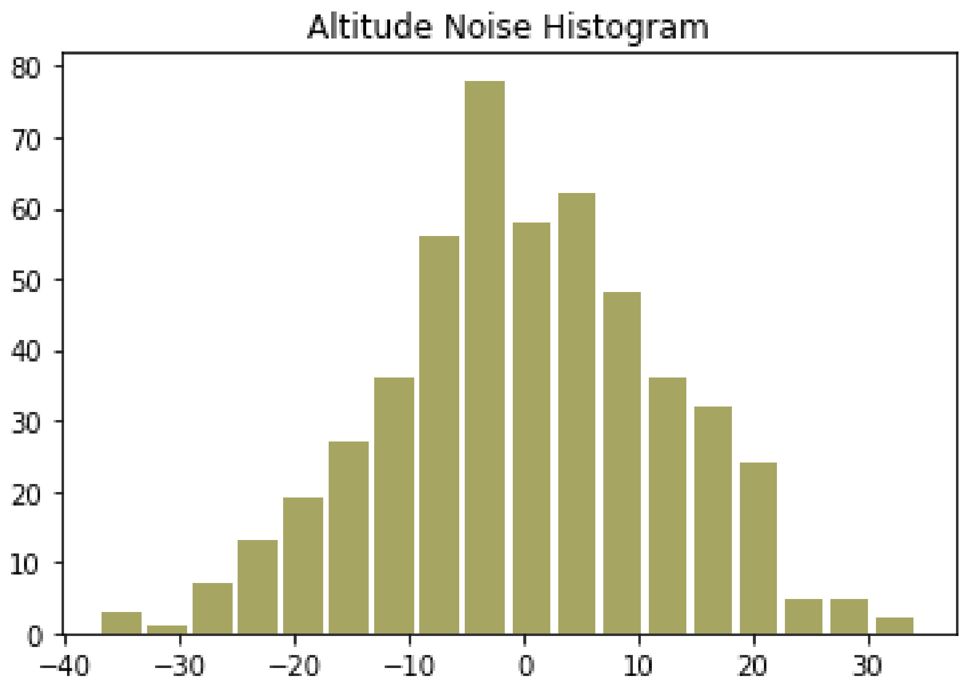
[11]:
```
zcm_variance = statistics.variance(zcm_detrended)
print("Altitude noise variance : {0} cm^2".format(zcm_variance))
```

```
Altitude noise variance : 149.95054299305454 cm^2
```

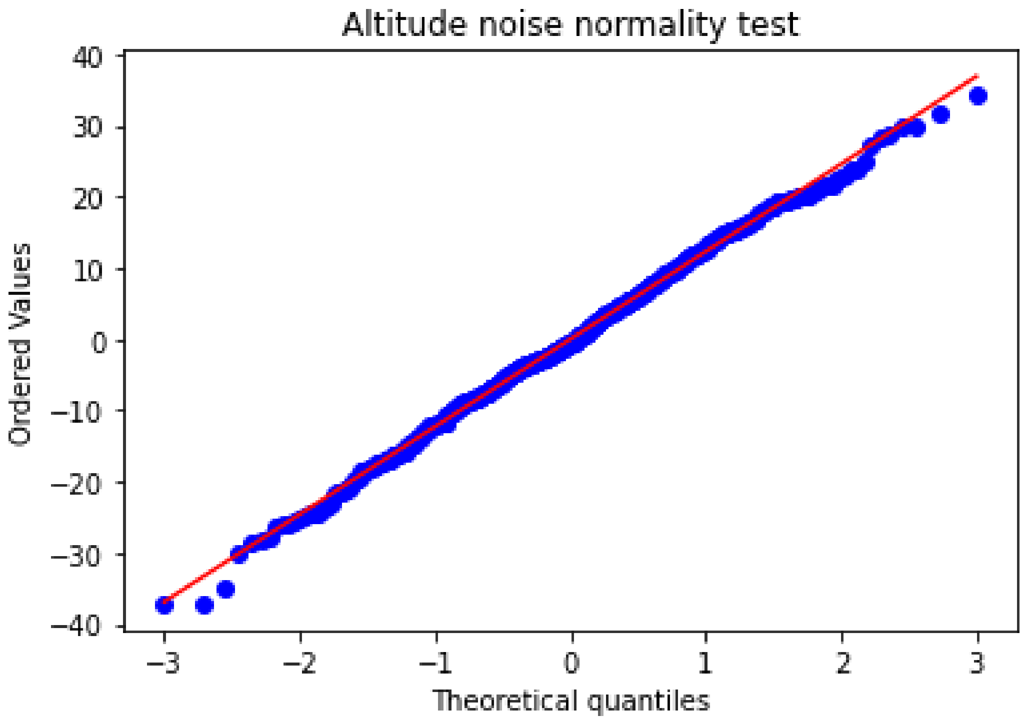
[12]:
```
statistic, p_value = normaltest(zcm_detrended)
print("Altitude noise normality test : p-value = {0}".format(p_value))
```

```
Altitude noise normality test : p-value = 0.6601968603417876
```

[13]:
```
n, bins, patches = plt.hist(x=zcm_detrended, bins='auto', color='#808020',
                            alpha=0.7, rwidth=0.85)
plt.title('Altitude Noise Histogram')
plt.show()
```



[14]:
```
probplot(x=zcm_detrended,dist='norm',plot=plt)
plt.title('Altitude noise normality test')
plt.show()
```


Altitude noise normality test
Ordered Values
Theoretical quantiles
40
30
20
10
0
−10
−20
−30
−40
−3
−2
−1
0
1
2
3

```
[ ]:
```